

REMARKS

Claims 1-10 are pending in the application and are rejected. The specification is objected to.

Specification

The specification is objected to for failing to provide antecedent basis for the terms "unit of processing" in claims 1, 2, 5, 7 and 10, and the term "processing-unit-by-processing-unit basis" in claim 6.

Although the applicants believe the specification does provide sufficient support for these terms (for example, see "units of processing" discussed on page 11 lines 13-14 and page 14 line 14), they nevertheless amend the claims as shown above and request reconsideration.

Claims

Claims 1-10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent 6,455,292 (referred to as "Oikawa") in view of Richard Stallman, "GNU Emacs 19 Manual" (referred to as "Stallman"). With regard to independent claims 1 and 6, the Office Action indicates Oikawa discloses all that is claimed except for the insertion of a user function based on a user-specified user analysis program between units of processing of a memory repair analysis program; that Stallman teaches the insertion of a user function based on a user-specified user analysis program between units of processing of a memory repair analysis program; and that it would have been obvious to combine these teachings for "allowing the user to customize behavior of the mode of the memory repair analysis program."

Applicants respectfully disagree with the Office Action's characterization of what is taught in Stallman. They also amend the claims as shown above and request reconsideration in view of those amendments and the following discussion.

Stallman pertains to a program called Emacs, which is a text editor. The portion of Stallman referred to by the Office Action describes a feature called a "hook" that may be used to customize the text editor. Although the Office Action does not explicitly refer to the hooks, its discussion appears to indicate that these hooks are deemed to correspond to the claimed user functions.

A recent version of the Stallman reference includes the following paragraph:

Hooks are an important mechanism for customization of Emacs. A hook is a Lisp variable which holds a list of functions, to be called on some well-defined occasion. (This is called running the hook.) The individual functions in the list are called the hook functions of the hook. With rare exceptions, hooks in Emacs are empty when Emacs starts up, so the only hook functions in any given hook are the ones you explicitly put there as customization.

The hooks in Stallman customize the operation of the text editor by programming or automating editing functions a user could perform. A user creates the hooks and instructs Emacs when to run them. With such customization, a user is spared the drudgery of manually setting options or performing various editing operations but Emacs remains a text editor. A user can edit text more efficiently but the hooks do not transform Emacs into a program with new capabilities that can process or manipulate new kinds of files with different structures.

In contrast to what is taught by Stallman, the claimed invention employs a user analysis program that is implemented by a set of user functions. These functions are inserted between operations of memory repair analysis in a semiconductor test apparatus. The user functions modify data as it passes from one operation to another. As a result, an existing analysis tool is transformed into something that can perform a type of analysis that it is otherwise incapable of performing. It continues to perform the operations suitable for semiconductors with a regular memory cell structure but it is "tricked" into doing what is necessary for a different type of memory structure.

Customization by a hook, for example, could adjust the amount that text is indented. By analogy, the claimed user functions could transform Emacs into a program that can edit files with a different structure such as bitmapped pictures. The Emacs editor would continue to operate as if it was processing a text file but user functions would modify data passed between its operations so that it actually processed bitmapped graphics. Applicants and the undersigned do not believe the "hooks" mentioned in Stallman are capable of providing this functionality. Even if they are, there is no suggestion in Stallman that they can be used in this manner.

Furthermore, the subject matter in Stallman differs greatly with semiconductor memory cell analysis and repair. Customizing a text editor, for example, is not very relevant to the analysis and repair of semiconductor memory. Even if Stallman did teach the same type of user function that is claimed, it would require inventive skill to recognize how these teachings for a text editor could be applied to semiconductor memory analysis and repair.

In other words, even if Stallman provides the teaching that is alleged by the Office Action, the skilled person would not know from Oikawa and Stallman how to modify the standard process to analyze and repair semiconductor memory with a new structure.

All of the remaining claims are dependent on either claim 1 or claim 6 and add further limitations.

With regard to claims 2 and 7, the Office Action acknowledges that Oikawa does not disclose the claimed limitations but it indicates Stallman does disclose them. With due respect, this simply is not correct. Stallman has nothing to do with memory analysis and repair.

With regard to claim 4, the Office Action indicates the tester-processor 20 in Oikawa corresponds to the claimed user analysis program storage and user function means. Applicants disagree. The tester-process (TP) 20 is merely a conventional processor that may correspond to the claimed "repair analysis computing unit" but applicants are unable to find anything that could correspond to the claimed user analysis program or user analysis means. If it is still believed there is such a correspondence, applicants request that the next communication clearly explain what aspect in Oikawa corresponds to the claimed user analysis program, which comprises user functions inserted as set forth in claim 1. (See comments below for claim 5.)

With regard to claim 5, the Office Action refers to element 10 in Fig. 1 as the claimed user analysis program storage section. Applicants disagree. Element 10 is a work station that has a repair-condition file but it does not have a user analysis program storage section.

With regard to claim 9, the Office Action relies on Stallman to teach something relevant to memory analysis and repair. Stallman does not teach anything in this field of endeavor.

With regard to claim 10, the Office Action indicates col. 8 lns. 28-36 and 52-56 in Oikawa teach the claimed limitations of "selecting a set of user functions that corresponds to the type of the semiconductor device under test from among a plurality of sets of user functions." Applicants are unable to find anything in the cited text that teaches or suggests this limitation.

CONCLUSION

Applicants amend the claims and request reconsideration in view of the preceding remarks.

Respectfully submitted,



David N. Lathrop
Reg. No. 34,655
601 California St., Suite 1111
San Francisco, CA 94108-2805
Telephone: (415) 989-8080
Facsimile: (415) 989-0910

Certificate of Transmission

I certify that this Response to Office Action and any following materials are being transmitted by facsimile on January 6, 2009 to the U.S. Patent and Trademark Office at telephone number (571) 273-8300.



David N. Lathrop